ก้องภพ มังคะตาระนา

SOFT MAIL

เขียนโปรแกรมให้เป็นโครงสร้างด้วยเทคนิคอย่างง่าย ๆ

รู้จักเทคนิคการเขียนโปรแกรมอีกสักนิด... อนาคตโปรแกรมเมอร์เมืองไทยคงไปได้สวย

สวัสดีครับเมื่อพูดถึงเรื่องนี้แล้วก็คงเป็นที่ทราบกันดีว่าคอมพิวเตอร์ได้เข้ามามีบทบาทในชีวิตประจำวันของเราในทุก ๆ ด้าน ไม่ว่าจะเป็นการจำหน่ายสินค้าก็มีคอมพิวเตอร์เข้ามาคิดราคาแทนคน, การคิดค่าน้ำค่าไฟต่อไปก็คงจะไม่มีคนมาเคาะประตูบ้าน ขออ่านมิเตอร์อีกต่อไปเพราะทางรัฐบาลจะจัดหาคอมพิวเตอร์มาอ่านมิเตอร์แทนคน (ผมก็คิดว่าคงดีมากเลยเพราะช่วยลดอัตราเสี่ยงของพนักงานอ่านมิเตอร์ต่อการถูกเจ้าด่างของท่านงับเอา) เงินเดือนค่าน้ำค่าไฟก็จ่ายผ่านทางบัญชี และยังมีอื่น ๆ อีกมากมายพูดได้ทั้งวันไม่จบไม่สิ้น

มาเข้าเรื่องกันดีกว่าเมื่อพูดถึงเรื่องข้างต้นแล้ว ก็ทำให้การใช้โปรแกรมสำเร็จรูปในคอมพิวเตอร์เพิ่มมากขึ้นตามไปด้วย เราก็ต้องมาเสียเงินเล็ก ๆ น้อย ๆ เป็นค่าโปรแกรมสำเร็จรูปอยู่เรื่อยไป แล้วทำไมเราไม่มาเขียนโปรแกรมไว้ใช้งานเองล่ะครับ พูดไปพูดมาผมก็วกเข้าเรื่องเดิมจนได้ "ไม่เอาแล้วผมพูดถึงการเขียนโปรแกรมกันดีกว่า"

โปรแกรมคืออะไร

โปรแกรมก็คือเครื่องมือทางความคิด ที่นำมาใช้ประมวลผลการทำงานทั้งยากและง่าย และเป็นตัวที่ช่วยจดจำขั้นตอนการทำงาน วิธีการคำนวณ คนเราจึงคิดว่าควรจะมีขั้นตอนทุกอย่างอย่างเป็นระเบียบและชัดเจน (การเขียนโปรแกรมที่มีขั้นตอนที่ไม่แน่นอนนั้นจะทำให้ยากแก่การตรวจสอบและแก้ไข)

ขั้นตอนในการเขียนโปรแกรม

การเขียนโปรแกรมที่ดีนั้น ผู้เขียนควรจะมีการเขียน การวิเคราะห์งานในส่วนทำงานแล้วนำมาเขียนโฟลว์ชาร์ตเพื่อเราจะง่ายในการสร้าง, การตรวจสอบและวิเคราะห์เพื่อพัฒนาต่อ ในอนาคตคนที่เขียนโปรแกรมขึ้นมาคงจะไม่อยากให้โปรแกรมที่เขียนขึ้นมาด้วยความยากลำบากต้องตายไปพร้อมกับเวลาที่ผ่านไปนะครับ (ผมหมายถึงเขียนโปรแกรมมาแล้วใช้ได้เพียงครั้งเดียว) ท่านอาจคิดว่าการเขียนการวิเคราะห์งานและโฟลว์ชาร์ตอาจจะดูเหมือนไม่มีความจำเป็นเท่าใดนัก แต่สำหรับโปรแกรมที่มีขนาดใหญ่หรือมีความซับซ้อน จะมีความจำเป็นอย่างมากเลยทีเดียว เหมือนกับเรามองจากด้านนอกเข้ามา ย่อมจะมองเห็นได้กว้างกว่า การมองจากด้านในออกไปในตอนแรก ๆ ผมก็ไม่ค่อยอยากจะเขียนโฟลว์ชาร์ตเอาเสียเลยเพราะผมมองว่ามันเป็นงานที่ซ้ำซ้อนทำให้เสียเวลา แต่จากที่ผมได้เขียนโปรแกรมโดยเริ่มจากการเขียนโฟลว์ชาร์ตก็ทำให้ผมสร้างงานได้เร็ว จะได้ประโยชน์จากโปรแกรมย่อยต่าง ๆ มากขึ้นและสามารถพัฒนาโปรแกรมต่อไปได้เรื่อย ๆ (การวิเคราะห์งานของโปรแกรมอาจจะไม่มีความจำเป็นในการเขียนโฟลว์ชาร์ตขนาดเล็ก) มาดูตัวอย่างกันบ้างดีกว่า

สมมติเราต้องการจะเขียนโปรแกรม UPDATE FILE เหตุมาจากผมแก้โปรแกรมไปเรื่อยเลยไม่อยากจะจำว่าได้แก้ไขไฟล์ใดไปบ้าง

การวิเคราะห์งาน

1. ส่วนที่ต่างกันดูได้จากเวลา คือไฟล์ทุกไฟล์ที่สร้างขึ้นใหม่หรือมีการแก้ไขจะมีเวลาเขียนไว้ที่ชื่อไฟล์ร่วมด้วยเสมอ

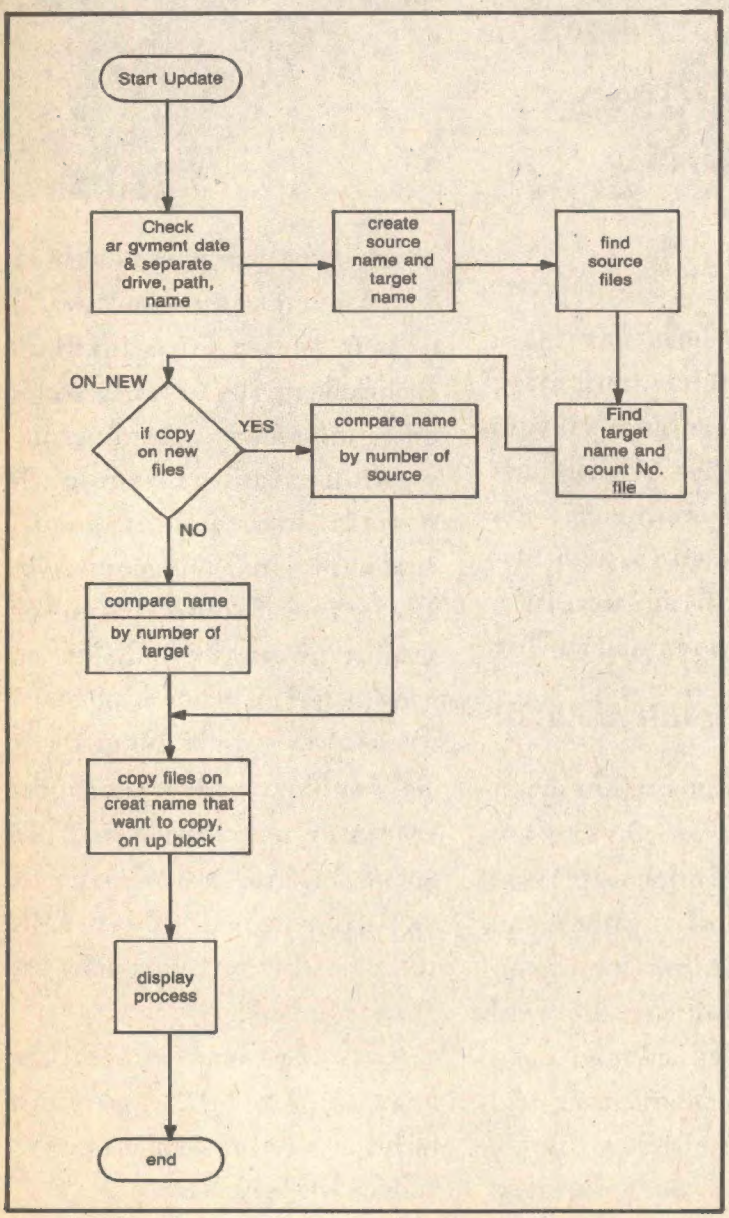
2. เราต้องอ่านชื่อไฟล์ใน source และ target ว่ามีไฟล์ใดอยู่บ้าง

3. นำมาเปรียบเทียบ วัน เวลาของไฟล์ในทั้งสองที่แล้วเก็บชื่อไฟล์ที่ใหม่แล้ว

4. อ่านข้อมูลจากไฟล์ที่ได้มาจากข้อ 2 และเขียนลงในที่เป้าหมายดังรูปที่ 1 และ 2

หลังจากที่เราเขียนโฟลว์ชาร์ตแล้ว เราก็จะมองออกว่าแต่ละขั้นตอนต้องการข้อมูลอย่างใดบ้าง ทำให้รู้ว่าต้องเขียนโปรแกรมแบบใด

โฟลว์ชาร์ตของทุกภาษาจะมีลักษณะที่เหมือนกัน เราจึงสามารถนำโฟลว์ชาร์ตไปเขียนเป็นภาษาอื่น ๆ ได้ ทุกชนิดแล้วเราจะปล่อยให้การเขียนโฟลว์ชาร์ตอยู่แต่ในความคิดของเราคนเดียวเท่านั้นหรือ เราควรจะแบ่งความคิดของเราให้คนอื่นเข้าใจบ้างเพื่อเขาจะได้สร้างเป็นภาษาของเขาเองบ้างเลยหรือ ผมพูดอย่างนี้อย่าเข้าใจผิดว่าผมมีผลประโยชน์อะไรกับบริษัทที่ผลิตอุปกรณ์ด้านนี้นะครับ ผมไม่ได้มีผลประโยชน์ใด ๆ เลยครับ



รูปที่ 1
อัลกอริทึม
MAIN PROGRAM

หลังจากที่เราได้โครงร่างเค้า ๆ และโฟลว์ชาร์ตของโปรแกรมแล้วเราก็นำมาเขียนเป็นโปรแกรมในภาษาที่เราต้องการ ภายในโปรแกรมเราก็ควรจะมีคำอธิบายของความหมายของคำสั่งนั้นๆ (comment) จะทำให้เราเข้าใจในประโยคส่วนนั้นได้ง่ายขึ้น เมื่อเราค้นหาเพื่อจะเพิ่มเติมหรือแก้ไขโปรแกรมก็ทำได้รวดเร็วและการพัฒนาโปรแกรมก็ทำได้รวดเร็วไปด้วย เท่าที่สังเกตมาก็พบว่าภาษาทุกอย่างมีคำอธิบายอยู่ภายในทุกภาษาอยู่แล้วยกเว้นอยู่ภาษาเดียวที่ไม่มีก็คือภาษาเครื่อง ผมอยากจะให้ทุกท่านใช้คำอธิบายความหมายให้ติดเป็นนิสัยเลยนะครับ และเมื่อท่านเขียนโปรแกรมจะหมดคำสั่งในชุดของรูปโฟลว์ชาร์ตนั้นแล้ว ก็วิเคราะห์ดูว่าได้ผลตามที่ต้องการหรือไม่ ถ้ายังไม่ได้ผลครบก็จัดการเขียนเพิ่มเติมต่อจนกว่าจะได้ผลตามที่โฟลว์ชาร์ตต้องการ

จากรูปที่ 1 เราก็จะต้องเขียนโปรแกรมดังนี้

1. เราจะต้องรับค่าจาก command line เข้ามาแล้วนำมาแยก ชื่อ, ที่อยู่, ช่องเก็บข้อมูล

2. สร้างชื่อที่อยู่ของ Target path โดยบวกชื่อ file เข้าไปด้วย (เพราะเราใส่มาเฉพาะที่อยู่)

3. หาไฟล์ที่อยู่ใน source

4. หาไฟล์ที่อยู่ใน Target

5. ตรวจดูว่ามีตัวเลือกให้ก๊อบปี้

ไฟล์ใหม่เข้ามาหรือไม่

6. มีคำสั่งก๊อปปี้ไฟล์ใหม่จากข้อ 5 เรียก function compare โดย compare file ที่อยู่ใน ที่อยู่ ต้นฉบับทั้งหมดกับไฟล์ในที่อยู่เป้าหมาย โดยดูจากวันที่เวลา โดยเลือกเอาเฉพาะไฟล์ใน source ที่ใหม่กว่า

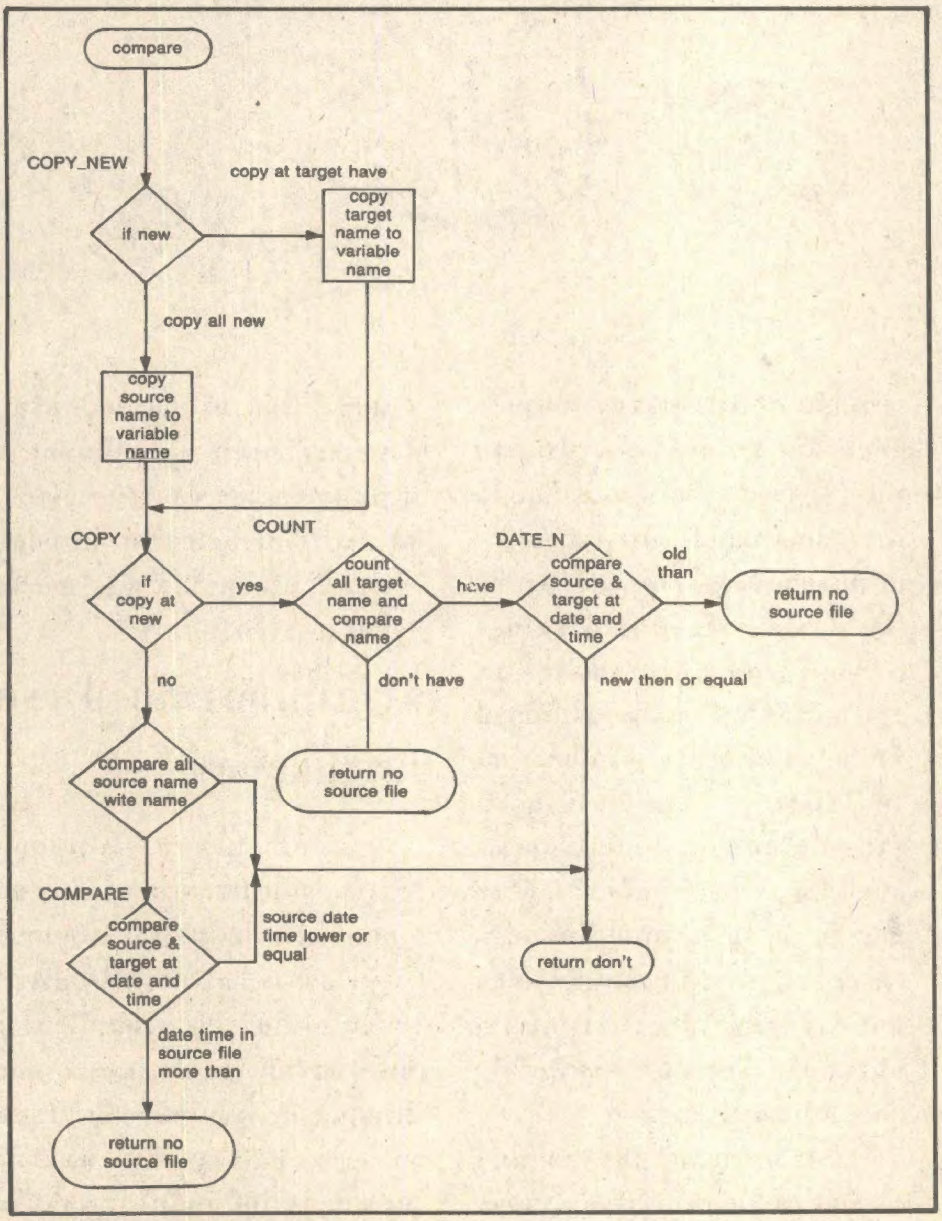
7. จากข้อ 5 ถ้าไม่มีตัวเลือกให้ก๊อบปี้ไฟล์ใหม่ก็จะเปรียบเทียบเฉพาะไฟล์ที่มีอยู่ในที่อยู่เป้าหมายเท่านั้น

8. Function copy files โดยจะลอกไฟล์ที่ได้มาจากการทำงานในข้อ 6 หรือข้อ 7

9. แสดงผลการทำงานของโปรแกรมทั้งหมด

การวิเคราะห์หาข้อผิดพลาด

การวิเคราะห์ข้อผิดพลาดนี้เราควรจะมีเครื่องมือช่วย เช่น เราอาจจะใช้โปรแกรม Debug ที่ให้มาพร้อมกับภาษานั้น ๆ เช่น ถ้าเราใช้เทอร์โบซีเขียน C ก็มี Turbo debuger คอยตรวจสอบความถูกต้องของโปรแกรม ถ้าไม่มีโปรแกรมดีบักมาด้วยเราก็สามารถสร้างดีบักด้วยตัวเราเอง โดยเราแทรกการพิมพ์ผลลัพธ์ของตัวแปรที่คิดว่าจะมีการผิดพลาดลงไประหว่างชั้นโฟลว์ชาร์ตต่าง ๆ เท่านี้เราก็จะได้ดีบักเกอร์ (debuger) เป็นของเราเอง เมื่อโปรแกรมที่สร้างถูกต้องแล้วเราก็มาทำอธิบายส่วนที่สร้างเพิ่มเติมที่ใช้ทำการตรวจสอบโปรแกรมที่เราเขียนเอาไว้ เพื่อจะได้เอามาตรวจสอบในคราวที่ต้องว่าการเปลี่ยนหรือเพิ่มเติมต่อไป ซึ่งไม่แน่ว่าจะมีการผิดพลาดเกิดขึ้นอีกหรือไม่



รูปที่ 2 อัลกอริทึมโพรซีเดอร์ Compare name

ประโยชน์

เรื่องนี้ผมเคยพูดเอาไว้บ้างแล้ว และเป็นจุดที่ผมจะเน้นมาก เรื่องการใช้โฟลว์ชาร์ตเขียนโปรแกรมเพราะว่ามันมีประโยชน์ ในการตรวจสอบ แก้ไข เพิ่มเติม และ พัฒนาต่อไป

การพัฒนาต่อไป

เรื่องนี้ผมก็เคยพูดไว้อย่างคร่าว ๆ เอาไว้แล้ว ผมก็จะขอพูดถึงรายละเอียดของการพัฒนาเลยก็แล้วกัน การปรับปรุงโปรแกรมเริ่มต้นด้วยการคิดว่าเราต้องการเพิ่มผลอะไรจากโปรแกรมบ้าง, ต่อไปก็มาดูที่โฟลว์ชาร์ตว่าตรงจุดใดเราควรจะเพิ่มอะไรเพื่อที่จะได้ผลตามที่ต้องการแล้ว เราก็จัดการเขียนส่วนนั้นลงไป แต่อย่าลืมนะครับว่าเราจะต้องเขียนส่วนที่เพิ่มลงไปในโฟลว์ชาร์ตด้วยเพื่อที่ได้การพัฒนางานในครั้งต่อไปได้ง่ายขึ้น สำหรับซอร์สโปรแกรมทางกองบรรณาธิการได้จัดทำเป็นซอฟต์เมล์บริการท่านผู้อ่านเช่นเคย!